

Notice of Allowability

Application No.

10/616,293

Examiner

Steven B. Theriault

Applicant(s)

SIMONYI, CHARLES

Art Unit

2179

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 08/29/2005.
2. ☒ The allowed claim(s) is/are 1-31 and 42-46.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some* c) ☐ None of the:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
- (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
- 1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
- (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☒ Other See Continuation Sheet.

BA HUYNH
PRIMARY EXAMINER

Continuation of Attachment(s) 9. Other: Fax from applicant - Proposed claim amendments.

EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Maurice Pirio on 11/04/2005.

In the Claims, please amend the claims as attached:(The application has been amended by applicant and sent to examiner via fax).

Allowable Subject Matter

1. Claims 1-31 and 42-46 are allowed.
2. The following is an examiner's statement of reasons for allowance:

Claims 1, 9, 15, 22, 28, and 42:

In the previous office action, the Examiner interpreted the claimed invention as a method of providing help information for program elements within program code in which the help information is provided to the user as the user selects the element from a tree representation of a program. The tree shows the ancestor path of extended classes and methods and therefore as the user selects each element they could visually see in the tree the ancestor structure. Help information is provided to the user, when requested, as an aid in determining the logical representation of the object as derived from root or program classes.

The closest prior art of Little et al. shows a program tree of element types with dependant information that can be selected by the user. The Examiner interpreted the tree of Little et al. as providing a hierarchical listing of objects (classes) and related methods within a package that represent a program. In viewing the classes and related methods within a tree the Examiner

interpreted Little et al. as providing an ancestral representation of an element type, for the purposes of displaying in a single location to a user a program and related components. Software tools such as the invention of Little attempt to solve a problem in the software development industry of simplifying, through an integrated development and modeling environment, the process of software development by modeling a program in a tree and providing a central tool repository with visual tools to develop program code.

Little et al. solves this problem by providing a development and modeling suite that automatically generates program code for the user as it is created during the design process and displays the program in a visually robust format. Little shows the program schema in the right window and shows the program elements in the left window in a tree, for the purposes of showing the entire program in one place. Little also shows the toolbars and tools for simplifying and aiding in the code creation process.

However, in light of Applicants arguments and upon reexamination of the prior art the Examiner withdraws the previous rejection, as the prior art of Little does not teach the ability to display the **ancestor chain for a given object comprising element types within a hierarchy** where the derivation is comprised of multiple program element types (See the present application specification page 5, Para 1, lines 12-16) along with help information for each type. The distinction of identifying and displaying an ancestor chain of multiple program element types differs from Little in that Little displays a structure of a program that shows relationships of elements organized in a tree with following structure: Package -> Class -> Method without ancestor information of the same program element type or help information. The prior art also does not teach where a given elements ancestors are shown in a progressively generic manner in the same tree or where the program element types are related to other program element types as Little organizes the information that is needed for the program and hides the abstraction of showing the root, and base classes that are inherently incorporated in any derived class.

Little also does not teach a method of providing help information for newly designed program element types with a computer program in which the **ancestor chain for a given object**

comprising element types within a hierarchy where the derivation is comprised of multiple program element types in which each element type is identified and displayed to the user on demand. While the prior art of Little will show the next class up in the chain, Little does not show the entire object chain up to the root along with help information in the display area. Further, the prior art does not teach where a user from the tree selects a new user defined method and the ancestor path for the newly created object is displayed to the root object. Most prior art systems have a recursive display processes that show the path one level at a time as the user selects them and most often does not display the root objects in the program window.

Finally, the prior art does not teach the displaying of help information where the element types are schemas that are hierarchically related in which the system automatically determines all of the ancestral objects within the schemas and displays identified ancestor program element type upon the user selecting a given element type from a program tree. For all the reasons listed above, the Examiner believes the present application in light of the current state of the prior art is in condition for allowance.

Claims 2-8, 10-14, 16-21, 23-27, 29-31, 43-46:

These claims are dependent upon Claims 1, 9, 15, 22, 28, and 42, respectively, and are thus allowable.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

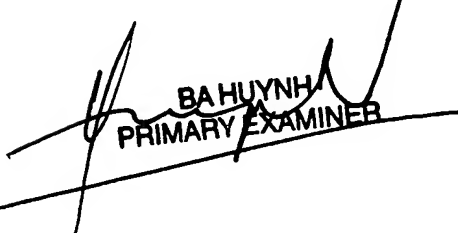
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Steven B. Theriault whose telephone number is (571) 272-5867. The examiner can normally be reached on M-F 7:30 - 4:00 PM.

Art Unit: 2179

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on (571) 272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

SBT


BA HUYNH
PRIMARY EXAMINER

Application No.: 10/616,293

Docket No.: 408068004US

PROPOSED AMENDMENTS TO THE CLAIMS

1. (Original) A method in a computer system for providing help information for a computer program, the method comprising:
providing a hierarchy of program element types, each program element type having associated help information;
providing a program tree representation of the computer program, the program tree having program elements, each program element having a program element type;
receiving from a user an indication to provide help information for a selected program element;
identifying ancestor program element types of the selected program element;
displaying to the user an indication of each identified ancestor program element type; and
upon receiving a selection of a displayed ancestor program element type, displaying to the user help information associated with the selected ancestor program element type.
2. (Original) The method of claim 1 wherein the displaying to the user of an indication of each ancestor program element type includes displaying naming information associated with a program element type.
3. (Original) The method of claim 1 wherein the help information associated with a program element type is stored as an attribute of a node representing the program element type.
4. (Original) The method of claim 1 wherein program element types are stored in schemas defining valid program trees at different levels of abstraction.

-1-

Application No.: 10/616,293

Docket No.: 408068004US

5. (Original) The method of claim 4 wherein a schema is represented as a tree data structure.
6. (Original) The method of claim 4 wherein each program element type is represented as a node within a tree.
7. (Original) The method of claim 1 wherein the ancestor program element types are specified by isa relationships, starting at the selected program element.
8. (Original) The method of claim 1 wherein an ancestor program element type includes a program element type defined in the program tree.
9. (Currently Amended) A method for providing help information for a computer program, the method comprising:
providing a specification of program element types, a program element type being defined as a specific instance derived from a more general program element type, program element types having associated help information;

~RV139.TXT

providing a computer program with program elements, each program element having a program element type;
receiving from a user an indication to provide help information for a selected program element;
identifying a derivation of program element types for the selected program element~ the derivation having multiple program element types;
displaying to the user an indication of program element types in the identified derivation; and
upon receiving a selection of a displayed program element type, displaying to the user help information associated with the selected program element type.

-2-

Application No.: 10/616,293

Docket No.: 408068004US

10. (Original) The method of claim 9 wherein the displaying to the user of an indication of each program element type includes displaying name information associated with a program element type.
11. (Original) The method of claim 9 wherein the help information associated with a program element type is stored as an attribute of a node representing the program element type.
12. (Original) The method of claim 9 wherein program element types are stored in schemas defining valid computer programs at different levels of abstraction.
13. (Original) The method of claim 9 wherein program element types are stored as definitions within the computer program.
14. (Original) The method of claim 9 wherein a program element type that is a specific instance of a more general program element type is defined by an is a relationship.
15. (Currently Amended) A method for providing information for a computer program, the method comprising:
providing a hierarchy of schemas defining valid computer programs, each schema specifying program element types at different levels of abstraction;
providing a computer program with program elements, each program element having a program element type;
identifying a derivation of program element types for a program element of the provided computer program from the provided hierarchy of schemas the derivation having multiple program element types; and
displaying an indication of program element types in the identified derivation.

-3-

Application No.: 10/616,293

Docket No.: 408068004U8

16. (Original) The method of claim 15 wherein the identifying of a derivation is performed in response to a user selecting a program element of the provided computer program.
17. (Original) The method of claim 15 including upon receiving a selection of a displayed program element type, displaying the information associated with the selected program element type.

18. (Original) The method of claim 15 wherein the displaying to the user of an indication of each program element type includes displaying name information associated with a program element type.
19. (Original) The method of claim 15 wherein the information associated with a program element type is stored as an attribute of the program element type.
20. (Original) The method of claim 15 wherein a program element type is stored as a definition within the computer program.
21. (Original) The method of claim 15 wherein a program element type that is a specific instance of a more abstract program element type is specified by an isa relationship.
22. (Currently Amended) A computer system for providing help information for a computer program, comprising:
 - a data structure storing a hierarchy of schemas defining computer programs, each schema specifying program element types of a computer program at different levels of abstraction;
 - a store within a computer program having program elements, each program element having a program element type;

-4-

Application No.: 10/616,293

Object No.: 408068004US

- a component that identifies a derivation of program element types for a program element of the computer program from the stored hierarchy of schemas the derivation having multiple program element types;
 - a component that displays an indication of program element types in the identified derivation; and
 - a component that displays help information associated with a program element type selected from the displayed indication of program element types.
23. (Original) The computer system of claim 22 wherein the identifying of a derivation is performed in response to a user selecting a program element of the computer program.
 24. (Original) The computer system of claim 22 wherein the displaying to the user of an indication of program element types includes displaying name information associated with a program element type.
 25. (Original) The computer system of claim 22 wherein the information associated with a program element type is stored as an attribute of the program element type.
 26. (Original) The computer system of claim 22 wherein a program element type is stored as a definition within the computer program.
 27. (Original) The computer system of claim 22 wherein a program element type that is a specific instance of a more abstract program element type is specified by an isa relationship.

-5-

Application No.: 10/616,293

Docket No.: 408068004US

28. (Currently Amended) A computer-readable medium containing instructions for controlling a computer system to provide information for a computer program, by a method comprising:
 - providing a computer program having program elements, each program element having a program element type, the program element types being defined by a hierarchy of schemas specifying program element types at different levels of abstraction;
 - identifying a derivation of program element types for a program element of the provided computer program from the hierarchy of schemas, the derivation havinn multintle program element types;
 - displaying an indication of program element types in the identified derivation; and
 - displaying help information associated with a program element type selected from the displayed indication of program element types.
29. (Original) The computer-readable medium of claim 28 wherein the identifying of a derivation is performed in response to a user selecting a program element of the computer program.
30. (Original) The computer-readable medium of claim 28 wherein the information associated with a program element type is stored as an attribute of the program element type.
31. (Original) The computer-readable medium of claim 28 wherein a program element type is stored as a definition within the computer program.
- 32-41. (Cancelled)

-6-

Application No.: 10/616,293

Docket No.: 408068004US

42. (Currently Amended) A method for providing help information for a computer program, the method comprising:
 - providing a program tree representation of the computer program, the program tree having program elements;
 - providing a hierarchical programmatic relationship for program elements of the program tree;
 - receiving from a user a selection of a program element of the program tree;
 - identifying a derivation of the provided hierarchical programmatic relationship for the identified program element, the derivation having multiple program elements, each program element having a program element type;
 - displaying to the user the identified derivation;
 - receiving from user a selection of programmatic relationship of the displayed derivation;
 - retrieving help information associated with the selected programmatic relationship;
 - and
 - displaying to the user the retrieved help information.
43. (Original) The method of claim 42 wherein the programmatic relationship is based on program trees representing the computer program with different levels of abstraction.

~RV139.TXT

44. (Original) The method of claim 42 wherein the programmatic relationship is based on the hierarchy of operators and operands in the program tree.
45. (Original) The method of claim 42 wherein the programmatic relationship is based on the organization of the computer program.
46. (Original) The method of claim 42 including identifying program elements related to the selected program element wherein the displaying includes displaying help information associated with the related program elements.

-7-